Run your own Bookstack docker

After being required to move my server and the LinuxServer BookStack image stopped working. At the same time I've been obligated to migrate my containers.

Delving deeper, I ended up making my own bookstack docker image - which turned out to be quite easy but not as easy as I had hoped. But I believe this path is the easiest to maintain and comprehend, in the end.

It's not "to the letter perfect" but these are the coarse steps to achieve the migratin. The Dockerfile is the largest hurdle, that is pasted below for your convenience.

What we have to achieve in this page:

- Create a container with MariaDB
- Create a docker image that has an installation of Bookstack
  - Requires PHP-FPM
  - Requires Lavarel
  - Requires GIT to pull the latest version

In the next page we will:

- Inspect the existing installation
- Copy the old database to the new one
  - Perform some maintenance tasks to update links in the old data
- Copy the old files to the new container

Backup your files

In the linuxserver there are some recusirve links to different folders, which makes it quite confusing. You could use "ls -l" to discover which are linked. There are all in /var/config somewhere.

Initially, you can start to discover and search for your data by opening an interactive shell onto the docker (this would be likely, to SSH into the QNAP NAS, and executing commands as such:)

```
docker ps
docker exec -i 87652832 /bin/bash
cd /var/config
ls -l
cd /var/www/public
ls -l
```

once you have found your files with going through your directories and are confident you've found your uploaded files:

```
docker ps
docker cp 9YI22323:/var/config/www/config/BOOKSTACK_APP_KEY ./BOOKSTACK_APP_KEY
docker cp 9YI22323:/var/config/www/images ./config/images
docker cp 9YI22323:/var/config/www/files ./config/files

docker cp  9YI22323:/var/www/.env ./.env
```

Create your dockerfile

Don't forget, that we are using a docker with linuxserver, and extracting the images and uploads. Then, recreating a fresh docker, and coying over the files for the image. In my case, I'm working in a folder with a subfolder WikiExport where my files reside. And the database already exists.

I copy them over, to test my docker image locally. And in a deployment stage, we will publish the image and bind these folders to the host. And link the .env file with the host.
So we can easily move them in the future or edit them without digging into the container!

For a new fresh docker, these lines should be uncommented:

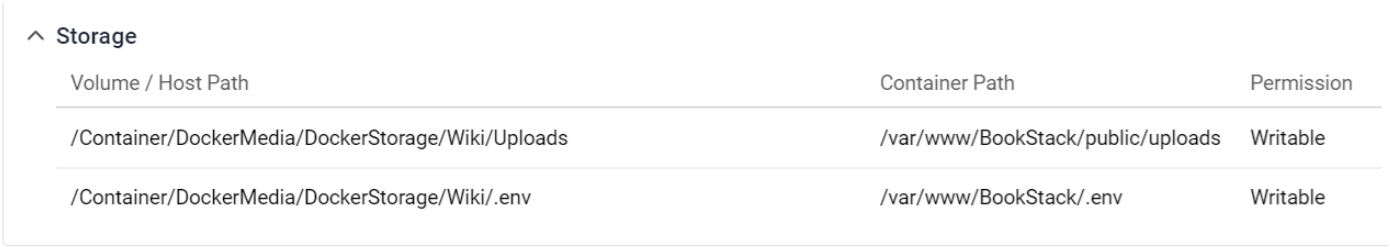
WORKDIR /var/www/BookStack
RUN php artisan key:generate
RUN php artisan migrate

```
FROM ubuntu:latest AS base

ENV DEBIAN_FRONTEND noninteractive
# from here https://dev.to/adnanbabakan/dockerizing-laravel-10-ubuntu-image-php-82-fpm-nginx-318p

# Install dependencies
RUN apt update
RUN apt install -y software-properties-common
RUN add-apt-repository -y ppa:ondrej/php
RUN apt update
RUN apt install -y php8.2\
    php8.2-cli\
    php8.2-common\
    php8.2-fpm\
    php8.2-mysql\
    php8.2-zip\
    php8.2-gd\
    php8.2-mbstring\
    php8.2-curl\
    php8.2-xml\
    php8.2-bcmath\
    php8.2-pdo

# Install php-fpm
RUN apt install -y php8.2-fpm php8.2-cli

# Install composer
RUN apt install -y curl
RUN curl -sS https://getcomposer.org/installer | php -- --install-dir=/usr/local/bin --filename=composer

# Install nodejs
RUN apt install -y ca-certificates gnupg
RUN mkdir -p /etc/apt/keyrings
RUN curl -fsSL https://deb.nodesource.com/gpgkey/nodesource-repo.gpg.key | gpg --dearmor -o
/etc/apt/keyrings/nodesource.gpg
ENV NODE_MAJOR 20
RUN echo "deb [signed-by=/etc/apt/keyrings/nodesource.gpg]
https://deb.nodesource.com/node_$NODE_MAJOR.x nodistro main" | tee /etc/apt/sources.list.d/nodesource.list
RUN apt update
RUN apt install -y nodejs
```

```
RUN apt install -y git
# Install nginx
RUN apt install -y nginx
RUN echo "\
    server {\n\
        listen 80;\n\
        listen [::]:80;\n\
        root /var/www/BookStack/public;\n\
        add_header X-Frame-Options \"SAMEORIGIN\";\n\
        add_header X-Content-Type-Options \"nosniff\";\n\
        index index.php;\n\
        charset utf-8;\n\
        location / {\n\
            try_files \$uri \$uri/ /index.php?\$query_string;\n\
        }\n\
        location = /favicon.ico { access_log off; log_not_found off; }\n\
        location = /robots.txt  { access_log off; log_not_found off; }\n\
        error_page 404 /index.php;\n\
        location ~ \.php$ {\n\
            fastcgi_pass unix:/run/php/php8.2-fpm.sock;\n\
            fastcgi_param SCRIPT_FILENAME \$realpath_root\$fastcgi_script_name;\n\
            include fastcgi_params;\n\
        }\n\
        location ~ /\.(?!well-known).* {\n\
            deny all;\n\
        }\n\
    }\n" > /etc/nginx/sites-available/default

RUN echo "\
    #!/bin/sh\n\
    echo \"Starting services...\"\n\
    service php8.2-fpm start\n\
    nginx -g \"daemon off;\" &\n\
    echo \"Ready.\"\n\
    tail -s 1 /var/log/nginx/*.log -f\n\
    " > /var/start.sh


RUN apt install -y vim
```

```
WORKDIR /var/www
RUN git clone https://github.com/BookStackApp/BookStack.git --branch release --single-branch
WORKDIR /var/www/BookStack

# RUN composer install
RUN composer install --no-dev
COPY ./.env .
COPY ./BOOKSTACK_APP_KEY.txt .
#COPY ./WikiExport/config/www/images /var/www/BookStack/public/uploads

# Set the bookstack folders and files to be owned by the user barry and have the group www-data
RUN chown -R www-data:www-data /var/www/BookStack

# Set all bookstack files and folders to be readable, writable & executable by the user (barry) and
# readable & executable by the group and everyone else
RUN chmod -R 755 /var/www/BookStack

# For the listed directories, grant the group (www-data) write-access
RUN chmod -R 775 /var/www/BookStack/storage /var/www/BookStack/bootstrap/cache
/var/www/BookStack/public/uploads

# Limit the .env file to only be readable by the user and group, and only writable by the user.
RUN chmod 640 /var/www/BookStack/.env

#RUN mkdir /config

#/var/www/BookStack/public/uploads#

COPY ./WikiExport/config/www/uploads/images /var/www/BookStack/public/uploads/images
COPY ./WikiExport/config/www/files /var/www/BookStack/public/uploads/files
#COPY ./WikiExport/app/www/



# WORKDIR /var/www/BookStack
# RUN php artisan key:generate
# RUN php artisan migrate
EXPOSE 80
CMD ["sh", "/var/start.sh"]
```

So we can run the new container on a host, with a bound path and manipulate the configuration or load/balance/mirror or backup our files without doing all this above work digging into the docker again.



Makig paths relative

To avoid some URL dependent code, you can open MySQL and execute simular queries, to change the absolute paths into relative paths.

```
SET SQL_SAFE_UPDATES = 0;
UPDATE  bookstackapp.images SET URL= REPLACE(REPLACE(url, 'http://wiki.sophior.com',
''),'https://wiki.sophior.com','')
UPDATE bookstackapp.settings set value = REPLACE(value, 'https://wiki.sophior.com', '') where value like
'%wiki%'
UPDATE bookstackapp.pages set html = replace(html, 'http://wiki.sophior.com', '') where HTML LIKE '%wiki%';
UPDATE bookstackapp.pages set html = replace(html, 'https://wiki.sophior.com', '') where HTML LIKE '%wiki%';
```

Mixed content error

I have found that it works quite okay if you run it via an internal IP and exposed port.

But if you have an NGINX in front, forwarding from a domain - you really should set your URLS rights.

You need to add to the .env file the line: APP_URL=HTTPS://xxxxxx to avoid mixed content. But from that point, you only can display the website via the external URL.

Revision #4
Created 8 October 2024 17:48:09 by Tim
Updated 21 October 2024 22:43:49 by Tim